CLAIMS

What is claimed is:

1. A method for restoring a memory value comprising:
identifying a first logic value stored in a first register;
branching to a first predefined location within programming code based upon the first logic value;
utilizing the first register as a scratch register during execution of the programming code;
restoring the first logic value back to the first register after execution of the programming code has finished.

2. The method of claim 1 further comprising detecting an occurrence of an interrupt during execution.

3. The method of claim 1 further comprising:

identifying a second logic value stored in a second register;

branching to a second predefined location within the programming code based upon the second logic value;

utilizing the second register as scratch register during execution of the programming code;

restoring the second logic value back to the second register in response to the second predefined location.

5. The method of claim 1, wherein the branching to a first predefined location further including identifying the first predefined location from a plurality of predefined locations in the programming code.

6. The method of claim 1 further comprising executing the programming code in a processor firmware layer.

7. The method of claim 1 further comprising storing the programming code in a non-volatile memory.

8. The method of claim 1 further comprising utilizing the first register as an index register during execution of the programming code.

9. The method of claim 1 further comprising utilizing the first register as a predicate register during execution of the programming code.

10. The method of claim 1 further comprising saving rest of processor states before execution of interrupt handlers.

11. A digital processing system comprising:
an execution unit;
a general purpose register file coupled to the execution unit containing a plurality of general-purpose registers;
a memory coupled to the execution unit for storing a processor abstraction layer, the processor abstraction layer further including interrupt handlers, each of the interrupt handler further including saving architecture state code, the saving architecture state code further including a plurality of predefined sections, wherein each the predefined section corresponds to a logic value of a register whereby the logic value can be restored in response to the predefined sections.

12. The digital processing system of claim 11, wherein the saving architecture state code is stored in a non-volatile memory.

13. The digital processing system of claim 11, wherein the register is multiple bits wide.

14. The digital processing system of claim 11, wherein the logic value can be restored in the register.

15. The digital processing system of claim 11, wherein the logic value is restored in a memory location.

16. The digital processing system of claim 11, wherein the register is a general-purpose register.

17. The digital processing system of claim 11, wherein the register is a predicate register.

18. An article of manufacture for use in a digital processing system for storing a logic value in a programming code, the article of manufacture comprising a machine readable medium having machine readable program code embodied in the medium, the program code comprising:

identifying a first logic value stored in a first register;

branching to a first predefined location within programming code in response to the first logic value;

utilizing the first register as a scratch register during execution of the programming code;

restoring the first logic value back to the first register in response to the first predefined location.

19. The article of manufacture of claim 18, further including computer readable program code for:

identifying a second logic value stored in a second register;

branching to a second predefined location within the programming code in response to the second logic value;

utilizing the second register as scratch register during execution of the programming code; and

restoring the second logic value back to the second register in response to the second predefined location.

20. The article of manufacture of claim 18, wherein the branching to a first predefined location further including computer readable program code for identifying the first predefined location from a plurality of predefined locations in the programming code.

21. The article of manufacture of claim 18, further including computer readable program code for executing the programming code in a processor firmware layer.

22. A computer system comprising:

means for identifying a first logic value stored in a first register;

means for branching to a first predefined location within a programming code based upon the first logic value;

means for utilizing the first register as a scratch register during execution of the programming code;

means for restoring the first logic value back to the first register in response to the first predefined location.

23. The computer system of claim 22, further comprising means for detecting an occurrence of an interrupt during execution.

24. The computer system of claim 22, further comprising:

means for identifying a second logic value stored in a second register;

means for branching to a second predefined location within the programming code in response to the second logic value;

means for utilizing the second register as scratch register during execution of the programming code;

means for restoring the second logic value back to the second register in response to the second predefined location.

25. The computer system of claim 22, wherein the branching to a first predefined location further including means for identifying the first predefined location from a plurality of predefined locations in the programming code.

26. The computer system of claim 22, further comprising means for executing the programming code in a processor firmware layer.

27. The computer system of claim 22, further comprising means for storing the programming code in a non-volatile memory.

28. The computer system of claim 22, further comprising means for utilizing the first register as an index register during execution of the programming code.

Add A1